

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-084960

(43)Date of publication of application : 31.03.1995

(51)Int.Cl.

G06F 15/00
G06F 1/00

(21)Application number : 06-187033

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 09.08.1994

(72)Inventor : CHAPMAN SYDNEY G
MICHAEL G TAYLOR

(30)Priority

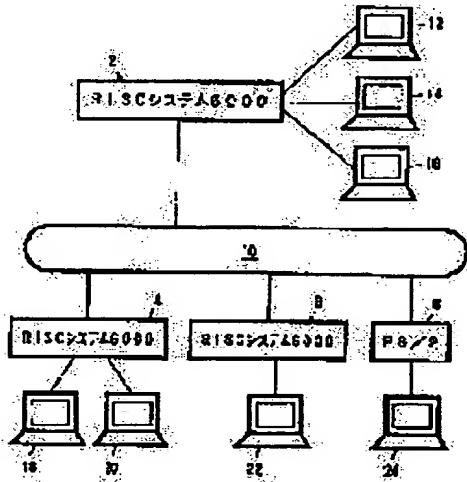
Priority number : 93 9318331 Priority date : 03.09.1993 Priority country : GB

(54) COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To provide concretely the temporary limit of access to such a system that has been made into a network.

CONSTITUTION: These computer system 2, 4, 6 and 8 connectable to plural users through the network are provided with a normal access control means for limiting user access to the systems 2, 4, 6 and 8 provided with a user confirmation procedure for comparing the identification of the user with the first definition of a permitted user, a system entire area profile means referred to by all the users of the systems 2, 4, 6 and 8 at the time of log-on and a temporary access control simplification means for temporarily preventing the access to the systems 2, 4, 6 and 8 by one or plural normal permitted users. The simplification means makes the privilege users of the systems 2, 4, 6 and 8 able to prepare the second definition of the users temporarily not permitted to be referred to by the system entire area profile means at the time of log-on.



(51) Int.Cl. ⁶ G06F 15/00 1/00	識別記号 330 370	府内整理番号 D 7459-5L E	F I	技術表示箇所
---	--------------------	--------------------------	-----	--------

審査請求 有 請求項の数 8 O L (全11頁)

(21)出願番号 特願平6-187033
 (22)出願日 平成6年(1994)8月9日
 (31)優先権主張番号 9318331.7
 (32)優先日 1993年9月3日
 (33)優先権主張国 イギリス (GB)

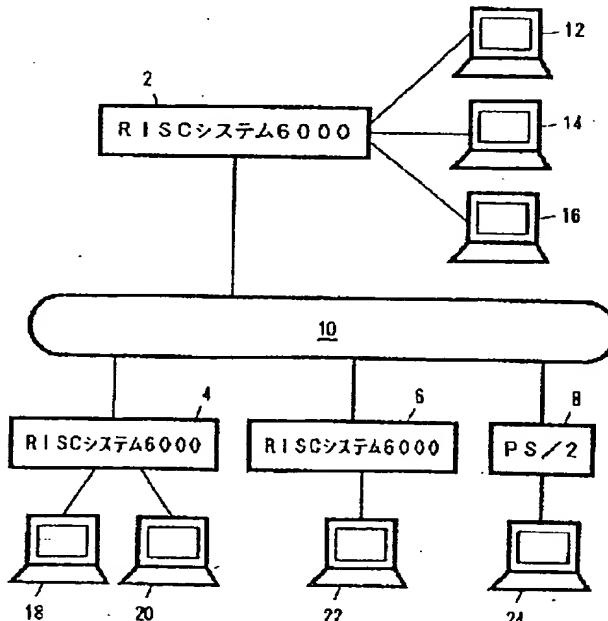
(71)出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレイション
 INTERNATIONAL BUSINESSES MACHINES CORPORATION
 アメリカ合衆国10504、ニューヨーク州
 アーモンク (番地なし)
 (72)発明者 シドニー・ジョージ・チャップマン
 イギリス エス021 2エイチ・ズイー
 ハンプシャー州ウィンチェスター サウス
 ウォンストン オークランド 2
 (74)代理人 弁理士 合田 潔 (外2名)
 最終頁に続く

(54)【発明の名称】コンピュータ・システムおよびアクセス防止方法

(57)【要約】

【目的】 本発明の目的は、ネットワーク化されたコンピュータ・システムに関し、具体的には、そのようなシステムへのアクセスの一時制限を提供することである。

【構成】 ネットワークを介して複数のユーザに接続可能なコンピュータ・システムに、ログオン時にユーザの識別を許可ユーザの第1定義と比較するユーザ確認手順を含む、システムへのユーザ・アクセスを制限するための、通常アクセス制御手段と、ログオン時にシステムの全ユーザによって参照される、システム全域プロファイル手段と、一人または複数の通常の許可ユーザによるシステムへのアクセスを一時的に防止するための一時アクセス制御簡便化手段とが含まれる。簡便化手段は、ログオン時にシステム全域プロファイル手段によって参照される、一時的に許可されていないユーザの第2定義を、システムの特権ユーザが作成できるようにする。



【特許請求の範囲】

【請求項1】ログオン時にユーザーの識別を許可ユーザーの第1定義と比較するユーザー確認手順を含む、システムへのユーザー・アクセスを制限するための、通常アクセス制御手段と、

ログオン時にシステムの全ユーザーによって参照される、システム全域プロファイル手段と、

ログオン時にシステム全域プロファイル手段によって参照される、一時的に許可されていないユーザーの第2定義を、システムの特権ユーザーが作成できるようにする、一人または複数の通常の許可ユーザーによるシステムへのアクセスを一時的に防止するための一時アクセス制御簡便化手段とを含む、ネットワークを介して複数のユーザーに接続可能なコンピュータ・システム。

【請求項2】簡便化手段が、システム全域プロファイルに一時的に許可されていないユーザーの第2定義を参照させるため、システム全域プロファイル手段を修正するように適合させていることを特徴とする、請求項1に記載のシステム。

【請求項3】簡便化手段が、システム全域プロファイルに、一時的にアクセスを拒否されるユーザーに一時アクセス制限に関する情報を提供させることを特徴とする、請求項1または請求項2に記載のシステム。

【請求項4】簡便化手段が、第2定義に従って一時的に許可されていない、すでにシステムにログ・オンしているユーザーをログ・オフさせる手段を含むことを特徴とする、請求項1ないし3のいずれかに記載のシステム。

【請求項5】簡便化手段が、猶予期間を定義し、すでにシステムにログ・オンしているユーザーに、猶予期間の満了時にシステム・アクセスが制限されることの警告を発行するように適合させていることを特徴とする、請求項4に記載のシステム。

【請求項6】簡便化手段が、指定された時刻に、一時的に許可されていないユーザーの一部またはすべてのアクセスを自動的に復元するように動作可能であることを特徴とする、請求項1ないし5のいずれかに記載のシステム。

【請求項7】システムがUNIXシステムであることを特徴とする、請求項1ないし6のいずれかに記載のシステム。

【請求項8】ログオン時にユーザーの識別を許可ユーザーの第1識別と比較するユーザー確認手順を含む、システムへのユーザー・アクセスを制限するための通常アクセス制御手段を有し、さらに、ログオン時にシステムの全ユーザーによって参照されるシステム全域プロファイル手段を有する、ネットワークを介して複数のユーザーと接続可能なシステムへのアクセスを一時的に防止する方法において、

システム全域プロファイル手段から一時的に許可されていないユーザーの第2定義を参照するステップと、

第2定義に従って一時的に許可されていないユーザーにアクセスを拒否するステップとを含む方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ネットワーク化されたコンピュータ・システムに関し、具体的には、そのようなシステムへのアクセスの一時制限に関する。

【0002】

【従来の技術】コンピュータ・ネットワークでは、単一のシステムが、ネットワークを介してそのシステムに接続された複数のユーザーに資源を供給できる。時折、そのシステムの負荷が高くなり過ぎ、一人または複数のユーザーが効率的に操作できなくなる場合があり、また、ある時間の間一部またはすべてのユーザーによるシステム資源へのアクセスを制限する必要が生じる場合がある。

【0003】複数ユーザー・システムの大半では、有効なユーザー名またはユーザー名とパスワードの有効な組合せをリストした1つまたは複数のファイルが保持され、ユーザーは、ログ・オン時にその名前とパスワードを供給することによってそのシステムへのアクセスを得る。

【0004】一時的にユーザーがシステム資源をアクセスできないようにするための従来技術の単純な方法の1つが、(必要な場合に)ユーザーをログ・オフさせ、そのユーザーのログオン・パスワードをリセットして、そのユーザーがアクセスを再取得できなくなることである。しかし、この方法は、管理と安全保護上の理由から、一般に望ましくない。多くのシステムでは、ユーザーの元のパスワードを再発行できなくなっている、その結果、そのユーザーがアクセスを再取得するためにはそのユーザーに新しいパスワードを通知する必要が生じる。

【0005】ユーザーがログ・オンできないようにするための従来技術のもう1つの方法が、許可ユーザーをリストしたファイルからそのユーザーのユーザー名を削除することである(これは、システムから1ユーザーを永久的に削除するための標準的な方法に類似しているが、永久削除の場合、さらに、ユーザーのデータ・ファイルが存在するならば、システム資源を解放するためにこれらのファイルも削除される)。通常のアクセスは、許可ユーザーのリストにそのユーザーの名前を復元することによって復元されるはずである。この方法の短所は、特に、ユーザー名を含むファイルに他のクリティカルなデータが含まれ、それらのデータも削除される場合に、データ消失の可能性があることである。

【0006】上の方法でアクセスの復元を簡単にするために、ログオン時に検査される活動コピー内のユーザーの項目を削除する前に、許可ユーザーをリストしたファイルをバックアップする(すなわち、安全な位置にコピーする)ことができる。システムを元の状態に復元するためには、バックアップしたファイルを、元の位置にコピーする。しかし、元のコピーが失われた場合、アクセス権

の復元が困難になる可能性がある。明らかに、ユーザ名やパスワードを含むファイルを変更する必要がないことが好ましい。

【0007】たとえば、UNIXオペレーティング・システムの一種であるIBM社のAIXオペレーティング・システムでは、"/etc/passwd"という名前のファイルに、ユーザ名と、暗号化されたパスワードを含む別のファイルの参照とが含まれる。このファイルがなくなると、そのシステムには「全く」ログオンできなくなる。回復するためには、オペレーティング・システムを導入し直す必要がある。UNIXオペレーティング・システムの詳細については、W.R.Stevens著"UNIX Network Programming", Prentice-Hall 1990を参照されたい。

【0008】この方法には他の欠陥もある。たとえば、アクセス制限の期間中に、新規ユーザの追加や既存ユーザの永久削除が行われると、活動状態のシステム・ファイルとそのバックアップの両方にこれを反映する必要が頻繁に発生するはずである。さらに、ログ・オンを試みた際にこの方法によってアクセスを一時的に拒否された通常の許可ユーザは、永久的に許可されていないユーザに提示される物と同一の「無効ログオン」メッセージを受け取るはずである。

【0009】システムによっては、ログオン・セッションではなく、特定のファイルへのアクセスを制限できるものがある。たとえばAIXの場合、ユーザが自分自身のデータ・ファイル（そのユーザの「ホーム・ディレクトリ」）をアクセスすることを一時的に禁止できる。これは、そのディレクトリに対する「アクセス・モード」を変更し、後程復元することによって実行できる。AIXの場合、これによってそのデータに対するすべてのユーザのアクセスが必ず除去され、第2のユーザが第1のユーザのディレクトリへのアクセスに頼っているならば、第2のユーザも所望のタスクを実行できなくなる可能性がある。さらに、データ・アクセスを拒否しても、第1のユーザが別の場所に記憶されたプログラムを実行することができ、したがってプロセッサ時間を占有できるので、必ず所望の結果を達成できるとは限らない。また、自分のファイルをアクセスしようとした際の第1のユーザへのメッセージは、役に立たないであろう。というのは、このメッセージによって、彼が自分のファイルをアクセスできないことは伝えられるが、一時アクセス制限が有効であることは伝えられないからである。

【0010】

【発明が解決しようとする課題】本発明の目的は、たとえばシステム・オーナーが資源集中タスクを完了できるようにするために、アクセスを一時的に制限でき、簡単に復元でき、従来技術の解決の短所を軽減するシステムを提供することである。

【0011】

【課題を解決するための手段】本発明によれば、この目

的是、ログオン時にユーザの識別を許可ユーザの第1定義と比較するユーザ確認手順を含む、システムへのユーザ・アクセスを制限するための、通常アクセス制御手段と、ログオン時にシステムの全ユーザによって参照される、システム全域プロファイル手段と、ログオン時にシステム全域プロファイル手段によって参照される、一時的に許可されていないユーザの第2定義を、システムの特権ユーザが作成できるようにする、一人または複数の通常の許可ユーザによるシステムへのアクセスを一時的に防止するための一時アクセス制御簡便化手段とを含む、ネットワークを介して複数のユーザに接続可能なコンピュータ・システムによって達成される。

【0012】本発明は、システム・オーナーなどの特権ユーザが、システムに対するユーザ・アクセスを制御できるようにし、その結果、必要な時に必要に応じてアクセスを制限できるようにする。特権ユーザだけが実行できるアクセス制御プログラムなどの一時アクセス制御簡便化手段によって、通常はシステムにアクセスできるが現在はログ・オフしているユーザが、一時的にアクセスを取得できないようにする。そのシステムにすでにログ・オンしている一時的に許可されていないユーザも、おそらくは猶予期間または警告の後に、ログ・オフさせることができる。しかし、現ユーザをログ・オフさせる手段は、本発明にとって本質的ではない。本発明を従来技術から区別する主要な特徴は、通常の許可ユーザの定義と別の一時的な許可ユーザの定義を設け、その結果、ユーザ・アカウント情報を修正する必要がなくなることである。

【0013】本発明によれば、特権ユーザが、一時的に、アクセスを部分的に制限するか、他のすべてのユーザによるアクセスを禁止できるようになり、この特権ユーザは、システムの性能と資源のすべてを使用できるようになる。

【0014】この解決策を用いると、システムのアクセスを簡単に復元でき、指定された時刻に自動的に復元することもできる。さらに、ユーザには、どのような理由で、誰によって、どれだけの時間の間アクセスを拒否されているかなど、適切な情報を与えることができる。

【0015】これより、UNIXシステムに関して本発明の好ましい実施例を説明する。UNIXは、複数ユーザ、多重タスク処理オペレーティング・システムであり、その流行は、強力で柔軟なインターフェースと、多数のベンダーのプラットフォームにまたがる標準化から生じたものである。UNIXのIBM版がAIXであるが、UNIXの標準化のおかげで、下記の説明の詳細の多くが、他の版のUNIXにも同様にあてはまる。

【0016】

【実施例】図1は、4つのシステム2、4、6および8がケーブル接続10によって互いに接続される、小規模で非常に単純なコンピュータ・ネットワークを示す図で

ある。これらのシステムは、リング構成で互いに接続されるものとして図示されているが、他の構成も同様に可能であり、周知であり、たとえばより遠隔地と接続できるようにするために、物理ケーブル接続を他のデータ伝送手段に置き換えることも可能である。システム2、4、6および8のそれぞれに、1つまたは複数のダム端末である端末12、14、16、18、20、22および24を接続するための備えがある。ユーザは、ある端末で、通常はネットワーク上の資源へのアクセスを得るためにユーザ名とパスワードを含むデータを入力する。図示の例では、システム2、4および6が、IBM社のRISCシステム/6000シリーズのシステムであり、第4のシステムであるシステム8が、IBM社のPS/2シリーズのシステムになっている。これらのシステムは、IBM社のAIXオペレーティング・システムを走行させるのに一般的に使用されるシステムである(AIX、RISCシステム/6000およびPS/2は、インターナショナル・ビジネス・マシーンズ・コーポレーションの商標である)。RISCシステム/6000システムは、比較的強力であり、複数のユーザによって使用することができる。たとえば、そのようなシステムの1つであるシステム2が、3台の直接接続(または「局所」)端末12、14および16を有し、さらに、ネットワークを介して接続される他の(「遠隔」)端末18、20、22および24を介してその資源を使用できるようにセットアップすることも可能である。PS/2であるシステム8は、通常は能力が劣り、端末24の直接接続された1ユーザのみによって使用される。このユーザは、ネットワーク上の他の資源を使用する権限を得ることができ、このPS/2のシステム8を、主に遠隔資源との通信の処理に使用することさえ可能である。本発明の目的のためには、システムの局所ユーザと遠隔ユーザを区別する必要がなく、その結果、「ネットワークを介する接続」は、両方のタイプの接続を含むと解釈される。

【0017】UNIXシステムでは一般に、ネットワークを介して1システムに接続された複数のユーザが、システムにログオンしてその資源を使用できる。時折、このためにシステムの負荷が重くなり過ぎ、システム・オーナー(局所ユーザである場合が多いが、遠隔ユーザである可能性もある)などの一人または複数のユーザが、効率的に操作できなくなる可能性がある。この、システム資源に対する需要を制限する能力のなさは、オーナーが資源集中タスクのためにシステムを使用したい時や、あるタスクの性能特性を正確に測定したい時に特に問題になる可能性がある。

【0018】したがって、システム・オーナーなどの特権ユーザが、ある時間の間システム資源に対する需要を制限できることが必要である。

【0019】どのUNIXシステムにも、「スーパーユーザ」という概念がある(スーパーユーザは、「スーパー

ユーザ権限」を有するとも称する)。スーパーユーザは、この実施例では特権ユーザである。スーパーユーザは、別のユーザが所有する資源(たとえばファイル)をアクセスできるという点で、通常ユーザより大きい権限を有する。スーパーユーザは、ユーザをシステムに追加でき、システムからユーザを削除でき、他のユーザのアクセス権を変更できる。また、スーパーユーザは、UNIXの"kill"コマンドを使用することによって、どのユーザが所有する「プロセス」(走行中のプログラム)であっても打ち切ることができる。これに対して通常のユーザは、自分自身のプロセスしかkillできない。

【0020】好ましい実施例の理解を助けるために、関連するUNIXシステムの特徴の一部を説明する。

【0021】UNIXシステムに新規ユーザを追加したい時には、「ユーザ・アカウント」を作成する。ユーザ・アカウントは、2つの部分から構成されるとみなすことができる。まず、どのUNIXシステムにも、ユーザ・アカウントとその特徴を定義するファイル、AIXの場合は/etc/passwdファイル30があり、その1項目によって、ユーザが、パスワードによる確認の後にシステムへのアクセスを許可される。次に、ユーザがそのユーザ自身のプログラムとデータを記憶することのできる、ユーザのホーム・ディレクトリがある。

【0022】図2に、3人のユーザ、Mike、SydおよびFredを含むAIXシステムの/etc/passwdファイル30の例を示す。/etc/passwdファイル30には、1ユーザごとに1レコードが含まれ、その構造は、次のとおりである。

【0023】最初のフィールドであるusername(ユーザ名)31は、'mike'など、ユーザのユニークな識別子または名前を示し、この名前を、ユーザがシステムに対して自分自身を識別するためにログオン時に供給する。第2のフィールドであるpassword(パスワード)32は、ログオン時にユーザを確認するためのフィールドである。IBM社のAIXバージョン3では、このフィールドに感嘆符が含まれ、これによって、別の(より安全な)ファイルにそのユーザの暗号化されたパスワードが含まれることが示される。UNIXの他の版やAIXの初期バージョンでは、このフィールドに実際に暗号化されたパスワードが含まれる。確認方法の正確な性質は、本発明には関係なく、多数の代替案が可能である。第3のフィールドであるuser number(ユーザ番号)33には、ユーザをさらに識別するユーザ番号が含まれる。これは、しばしば(必ずではない)ユーザごとに独自である。AIXの場合、スーパーユーザは、0のユーザ番号によって表され、複数のスーパーユーザが存在してよい。第4のフィールド34は、そのユーザが属するグループを識別する番号である。通常、複数のユーザがこのフィールドに同一の項目を有するが、これは、たとえば部署番号とすることができる。第5のフィールド35は、テキスト・フ

フィールドであり、その内容は本質的でない。このフィールドは、たとえば、システム管理者がユーザーの完全な氏名を示すのに使用することができる。第6のフィールドであるhome directory (ホーム・ディレクトリ) 36は、ログオンが成功裡に完了した時にそのユーザーが居るディレクトリである、そのユーザーのホーム・ディレクトリを指定する。最後の第7のフィールドであるinitial program (初期プログラム) 37には、初期プログラムまたは「シェル」が含まれる。ユーザーの初期プログラムは、ユーザーがログ・オンした時、ユーザーの環境を初期設定し、ホーム・ディレクトリに移動した後に走行する (UNIXの場合、シェルは、コマンド・インターフリタであり、ユーザーが対話式ログオン・セッションを有するために走行していかなければならないプログラムである)。ユーザーの初期プログラムに関しては、ログオン手順の説明で解説する。

【0024】また、どのUNIXシステムにも、システムの全ユーザーの各ログオン・セッションの間に走行し、したがって、「システム全域」と表現することのできる初期設定手順がある。IBM社のAIXオペレーティング・システムの場合、このプログラムは、/etc/profileという名前のファイルに記憶されている。これは、システム・オーナーや管理者などの特権ユーザーが、ログオン時にすべてのユーザーに提供される追加機能を指定するために設けられたファイルである。このファイルが事故によって失われたり、修正中に損傷を受けても、通常は悲惨な結果にはならない (たとえば、ログオンできなくなったりはしない)。しかし、通常は、多くのユーザーがこのファイルをアクセスする権限を有することは望ましくない。というのは、このファイルを修正することによって、他のユーザーのデータを破壊できるからである。このシステム全域プロファイルを修正するには、通常はスーパーユーザであるか、スーパーユーザ権限を有する必要がある。

【0025】本発明の好ましい実施例では、一時的に許可されていないユーザーが、ログオンを試みた時にアクセスを拒否されるようにし、そのユーザーにアクセス制限に関する情報を与えるメッセージを表示できるようにするコードが、システム全域プロファイルに追加される。

【0026】UNIXのログオン・シーケンス UNIXのログオン・シーケンスについてさらに説明すれば、本発明の好ましい実施例の理解に役立つであろう。この説明は、AIXのログオン・シーケンスに基づくものであり、図3がその概略図である。

【0027】活動状態のUNIXシステムであるシステム2に直接接続された端末12、14および16は、通常は、ログイン・プロンプトを有する。また、遠隔端末である端末18、20、22および24の場合には、共通して理解されるプロトコルを使用して、遠隔システムであるシステム2に関するセッションをオープンし、ログイン・プロンプトを取得することができる。AIXの場

合、ユーザーがログイン・プロンプトに対して文字列を入力した時に、ステップ40で"login" (ログイン) コマンドが起動され、このコマンドが、ステップ42でアカウントの詳細を検査し、ステップ50でユーザーの環境をセットアップし、ステップ60でユーザーをそのホーム・ディレクトリに置き、ステップ62でユーザーの初期プログラムを始動し、ステップ70でプロファイルを走らせ、ステップ80でプロンプトを表示し、この時点でユーザーがコマンドを入力できるようになる。これらのステップを、以下で詳細に説明する。

【0028】ステップ42のアカウント詳細の検査には、/etc/passwdファイル30を参照して、アクセス取得を試みているユーザーの供給したユーザー名と一致するユーザー名31が存在するかどうかを検査する、ユーザーのアカウントを検査するステップ44、検査済みのユーザー名31に対応する暗号化された真のパスワード32を、アクセス取得を試みているユーザーの供給したパスワードを暗号化したものと比較することによってユーザーの識別を検査する、ユーザーを確認するステップ46、および、ユーザー・データベースに記憶され、ユーザーの責任能力とシステム2上のファイルに対するアクセス権を定義する、信用証明を確立するステップ48が含まれる。

【0029】したがって、アカウント詳細を検査するステップ42は、通常は、あるユーザーがシステム2へのアクセスを許可され、ログオン・シーケンスの後のステップに進めるか否かを判定するステップである。下で述べるように、本発明の好ましい実施例では、この後の段階でユーザーのアクセスを拒否できる。

【0030】ユーザーの環境をセットアップするステップ50は、本発明には特に関係ない。簡単に述べると、このステップは、ユーザー・データベースからユーザー環境を初期設定するステップ52と、/etc/environmentという名前の構成ファイルからユーザー環境を初期設定するステップ54からなる。これによって、ユーザーにそのユーザーの好みの、ある個数のシステム変数によって指定される環境が提供される。たとえば、ある変数が、そのユーザーの言語を決定する。

【0031】その後、ステップ60で、ユーザーが、/etc/passwdファイル30内でそのユーザーのホーム・ディレクトリとして指定されたホーム・ディレクトリ36に置かれ、ステップ62で、そのディレクトリ内から、やはり/etc/passwdファイル30で指定されるそのユーザーの初期プログラム37を走らせる。AIXの場合、初期プログラム37は、ファイル/etc/security/login.cfgの"shells"にリストされたプログラムのいずれかに制限される。初期プログラム37は、ユーザーにコマンド行インターフェースを提供し、このインターフェースが、ユーザーの入力したコマンドを解釈する。次に、ステップ70で、プロファイルと称するプログラムを走らせる。まず、ステップ72で、前に述べたシステム全域プロファ

イルである/etc/profileを走らせる。そのユーザのホーム・ディレクトリ36内に.profileという名前のファイルがある場合、ステップ74でそれを走らせる。このファイルには、ユーザ自身が指定でき、各ログオンの間に提供されることを望む機能が含まれる。

【0032】最後に、ステップ80で、「シェル・プロンプト」を提供し、ここでユーザがコマンドを入力できるようになる。これで、図3のAIXログオン・シーケンスの説明を終える。

【0033】アクセスの一時制限

必要な背景を提供し終えたので、本発明の好ましい実施例の方法を、図4の概略図を参照しながらAIXシステムの場合に関して詳細に説明する。

【0034】これらのステップは、マウスまたは他の適当な手段によって、1つまたは複数のメニュー画面上で、たとえば下記のようにコマンド行またはプロンプトで、特権ユーザが必要なデータを入力できるようにし、アクセスを制限しようとしているシステム上で走行する、アクセス制御プログラムの制御下で実行されることが好ましい。

【0035】この方法は、下記の諸ステップからなる。

【0036】1)ステップ81で、アクセス制御プログラムを呼び出す。ここでは、アクセス制御プログラムが、複数のパラメータまたは引数を有するコマンドの入力によって始動されると仮定する。

【0037】2)ステップ82で、このプログラムのユーザが、その権限を与えられていることを検査し、そうでなければステップ83で脱出する。

【0038】UNIX用語では、特権ユーザはスーパーユーザ権限を有する必要がある。AIXの場合、ユニークなユーザ番号33が0の場合がこれにあてはまる。複数のユーザがスーパーユーザ権限を有することも可能である。

【0039】3)ステップ84で、供給されたコマンド行引数を整列し、

- a) アクセスを許可されるユーザと、
- b) アクセス制限の持続時間（不定でもよい）と、
- c) ユーザが作業を完了するための猶予期間と、
- d) アクセスを制限するのか復元するのかと

を示すパラメータをセットし、明示的に供給されなかつたパラメータや無効にされるパラメータについては省略時値をセットする。たとえば、最少の猶予期間またはアクセス制限の最大の持続時間を確保する省略時値が望ましい場合がある。

【0040】4)ステップ85で、d)を検査して、アクセスを制限するのか、前のアクセス制限に従ってアクセスを復元するのかを確定する。アクセスを復元する場合については、後で説明する。

【0041】5)ステップ86で、c)を検査して、猶予期間が定義されているかどうかを確定し、定義されている場合、ステップ87で、差し迫ったアクセス制限に

ついてユーザに通知する警告をユーザに発行し、猶予期間だけ待つ。

【0042】この通知には、アクセス制限を開始した特権ユーザの名前、影響を受けるユーザ、猶予期間、アクセス制限の持続時間または制限の理由など、プログラムから入手可能か特権ユーザが供給する有用な情報のどれでも含めることができる。

【0043】6)ステップ88で、システム全域プロファイル(/etc/profile)のバックアップ・コピーを作

10

【0044】7)ステップ89で、a)を検査し、適当な方法によって、一時的に許可されていないユーザの定義を作成する。

【0045】たとえば、一時的に許可されていないユーザまたは一時的な許可ユーザのユーザ名のリストを作成できるはずであり（ログオン・シーケンスのユーザのアカウントを検査するステップ44によって、無効なユーザ名が認められないことが保証されるので、後者のリストには無効なユーザ名を含めることも可能である）、さもなくば、ユーザ番号33が指定された範囲内にあることを要求することも可能である。このプログラムを実行している特権ユーザは、必ず含まれるはずである。

【0046】8)ステップ90で、システム全域プロファイル/etc/profileを修正する。

【0047】この修正には、ログ・オン中のユーザが定義によって一時的に許可されなくなるかどうかを検査するステップと、そうである場合に、メッセージを表示し、そのユーザをログ・オフさせる（次のステップで説明する“kill”コマンドを使用する）ステップをもたらすコードを、/etc/profileに追加することが含まれる可能性がある。明らかに、/etc/profileは、空であったり必ずしも存在するとは限らない定義を探索するように永久的に適合させることもできる。

【0048】システム全域プロファイルは、ログオン時にすべてのユーザが参照するので、このような修正によって、現在そのシステムからログ・オフしている一時に許可されていないユーザは、アクセスを取得できなくなる。

【0049】9)すでにシステムにログ・オンしている一時に許可されていないユーザをログ・オフさせる。

【0050】AIXでは、一人のユーザが、1つまたは複数の「プロセス」またはプログラムを同時に走行させることができ、そのそれぞれが、ユーザ名31とユーザ番号33に関連するプロセス識別番号(PID)を有する。ユーザのログ・オフには、そのユーザのプロセスのすべてを打ち切ることが含まれる。“ps”コマンドは、既存のプロセスとそれに関連するユーザ名31とユーザ番号33をリストするコマンドである。したがって、一時に許可されないと定義されたユーザの所有するプロセスを識別できる。打切りは、“kill”コマンドにPIDを渡すこ

50

とによって達成される。前に述べたように、スーパーユーザ権限を有するユーザだけが、他のユーザのプロセスをkillできる。

【0051】10) ステップ92で、b) を検査して、アクセス制限に関して一定の持続時間が指定されたかどうかを確定する。一定の持続時間が指定されている場合、ステップ94で、バッチ・ジョブを実行要求して、適当な時刻にアクセスを復元し(AIXの"at"コマンドを使用する)、ステップ95で、アクセス制御プログラムから脱出する。持続時間が不定の場合、ステップ93で、バッチ・ジョブの実行要求を行わずにアクセス制御プログラムから脱出する。

【0052】アクセスの復元

一時的制限の後にシステムへのアクセスを復元するためには、ログオンを試みる通常の許可ユーザに対するアクセス拒否を、システム全域プロファイルがそれ以降に引き起こさないことを保証する必要がある。これによって、アクセス制御プログラムによってログ・オフされたユーザが、アクセスを再取得できることも保証される。

【0053】復元を最も簡単に行うには、ステップ90のシステム全域プロファイルの修正を逆転する、すなわち、修正されたプロファイルをステップ88で作ったバックアップ・コピーに置換する。その代わりに、一時的に許可されていないユーザの定義を修正または削除するか、システム全域プロファイルがその定義をアクセスできないようにすることも可能である。望むならば、一時的に許可されていないユーザのより小さな集合を指定するように定義を修正することによって、アクセスを部分的に復元することもできる。

【0054】この実施例では、ステップ81で、特権ユーザが"restore"（復元）パラメータを用いてアクセス制御プログラムをもう一度呼び出すことができ、この場合、ステップ96に進んで、修正されたシステム全域プロファイルをきれいなバックアップ・コピーと置換する。アクセスの制限に関して一定の時間が指定されていた場合には、特権ユーザがもう一度介入する必要はない。ステップ94で、アクセス制御プログラムが、バッチ・ジョブを実行依頼して、指定された時刻にシステム全域プロファイルを自動的に置換するからである。

【0055】あるユーザの初期プログラム37を、そのユーザをログ・オフさせるプログラム（これは/etc/security/login.cfgにリストされなければならない）に置換することによっても、ユーザのアクセスを拒否できることに留意されたい。しかし、これは望ましくないと思われる。というのは、ユーザの初期プログラムは、通常はシステム全域ではないからであり、また、この方法では、クリティカルな/etc/passwdファイルを修正するか、たとえば/bin/kshの内容をそのユーザをログ・オフさせるプログラムと交換する必要があるからだ。しかし、この方法によれば、より有用なメッセージを与える

ことができるはずである。

【0056】本明細書に記載の好ましい実施例では、アクセス制御プログラムを使用して必要な動作を実行するが、当業者であれば他の代替案を想像できるであろう。たとえば、アクセスの制限と復元に別々のプログラムを使用することが可能であり、実際に特権ユーザがこの動作の一部を手動で実行することも可能である。他の代替案には、一時的に許可されていないユーザの単一の定義を、2つの定義すなわち、現在ログ・オンしているユーザ用とすでにログ・オフしているユーザ用の2つに交換することが含まれるであろう。

【0057】また、諸機能を1つの大きなプロファイルに組み込むのではなく、一時アクセス制限だけのために別のシステム全域プロファイルを設け、所望の場合にそのプロファイルを通常のアクセス制御手順より優先させることも可能である。

【0058】さらに、UNIXシステムに関して好ましい実施例を説明してきたが、通常の許可ユーザの定義を参照するアクセス制御手段によってユーザ・アクセスが普通に制限され、通常のアクセス制御手段を満足するすべてのユーザが、少なくともログオン時に共用プロファイルまたは共通プロファイルを参照できるシステムであれば、どのシステムでも一時アクセス制御手段を設けることができる。

【0059】まとめとして、本発明の構成に関して以下の事項を開示する。

【0060】(1) ログオン時にユーザの識別を許可ユーザの第1定義と比較するユーザ確認手順を含む、システムへのユーザ・アクセスを制限するための、通常アクセス制御手段と、ログオン時にシステムの全ユーザによって参照される、システム全域プロファイル手段と、ログオン時にシステム全域プロファイル手段によって参照される、一時的に許可されていないユーザの第2定義を、システムの特権ユーザが作成できるようにする、一人または複数の通常の許可ユーザによるシステムへのアクセスを一時的に防止するための一時アクセス制御簡便化手段とを含む、ネットワークを介して複数のユーザに接続可能なコンピュータ・システム。

(2) 簡便化手段が、システム全域プロファイルに一時的に許可されていないユーザの第2定義を参照させるため、システム全域プロファイル手段を修正するように適合させていることを特徴とする、上記(1)に記載のシステム。

(3) 簡便化手段が、システム全域プロファイルに、一時的にアクセスを拒否されるユーザに一時アクセス制限に関する情報を提供させることを特徴とする、上記

(1) または(2)に記載のシステム。

(4) 簡便化手段が、第2定義に従って一時的に許可されていない、すでにシステムにログ・オンしているユーザをログ・オフさせる手段を含むことを特徴とする、上

記（1）ないし（3）のいずれかに記載のシステム。

（5）簡便化手段が、猶予期間を定義し、すでにシステムにログ・オンしているユーザに、猶予期間の満了時にシステム・アクセスが制限されることの警告を発行するよう適合させていることを特徴とする、上記（4）に記載のシステム。

（6）簡便化手段が、指定された時刻に、一時的に許可されていないユーザの一部またはすべてのアクセスを自動的に復元するように動作可能であることを特徴とする、上記（1）ないし（5）のいずれかに記載のシステム。

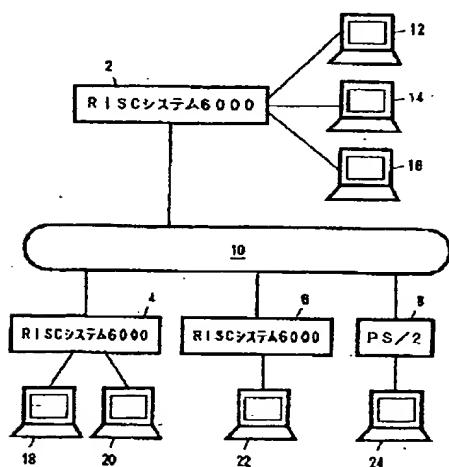
（7）システムがUNIXシステムであることを特徴とする、上記（1）ないし（6）のいずれかに記載のシステム。

（8）ログオン時にユーザの識別を許可ユーザの第1識別と比較するユーザ確認手順を含む、システムへのユーザ・アクセスを制限するための通常アクセス制御手段を有し、さらに、ログオン時にシステムの全ユーザによって参照されるシステム全域プロファイル手段を有する、ネットワークを介して複数のユーザと接続可能なシステムへのアクセスを一時的に防止する方法において、システム全域プロファイル手段から一時的に許可されていないユーザの第2定義を参照するステップと、第2定義に従って一時的に許可されていないユーザにアクセスを拒否するステップとを含む方法。

【0061】

【発明の効果】本発明により、アクセスを一時的に制限でき、簡単に復元でき、従来技術の短所を軽減するシステムを提供することができる。

【図1】



【図2】

31 32 33 34 35 36 37
 mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh
 syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh
 fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

30

31 32 33 34 35 36 37

mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

syd:!: 253:100: Syd Chapman : /u/syd : /bin/csh

fred:!: 125:101: Fred Bloggs : /u/fred : /bin/ksh

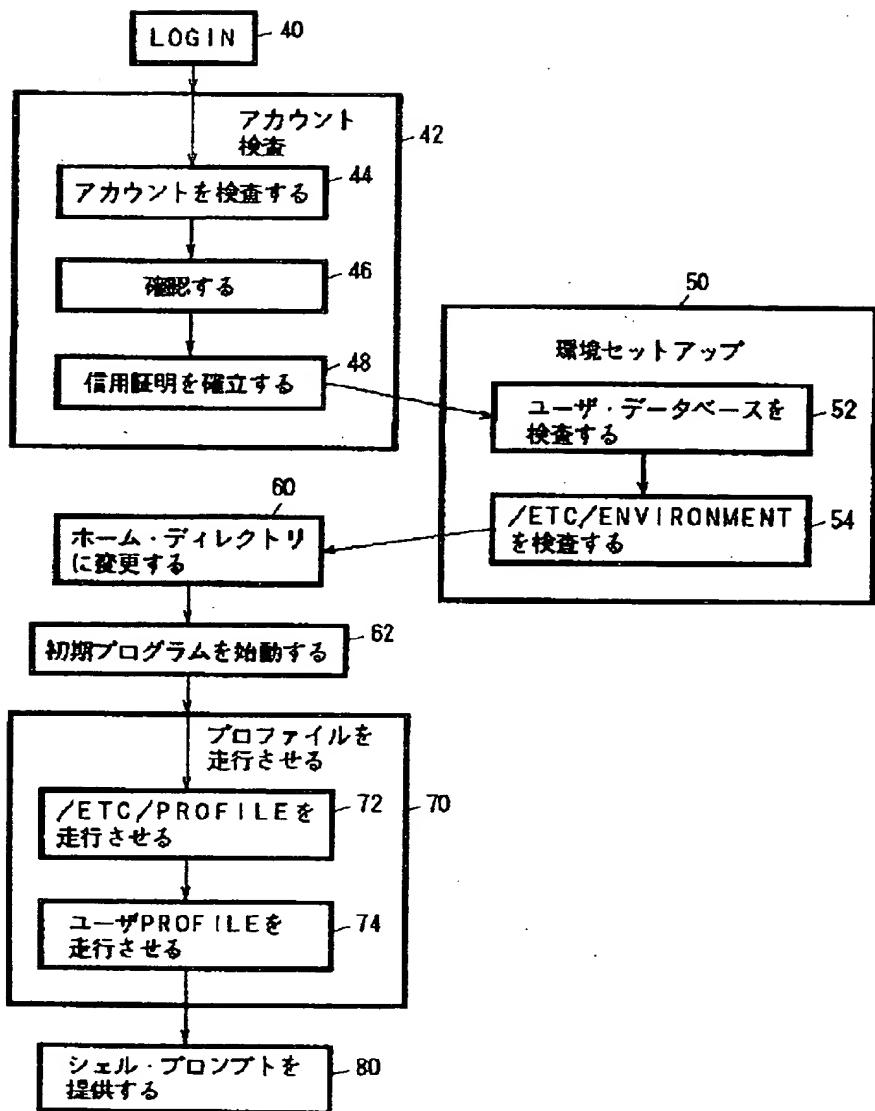
30

31 32 33 34 35 36 37

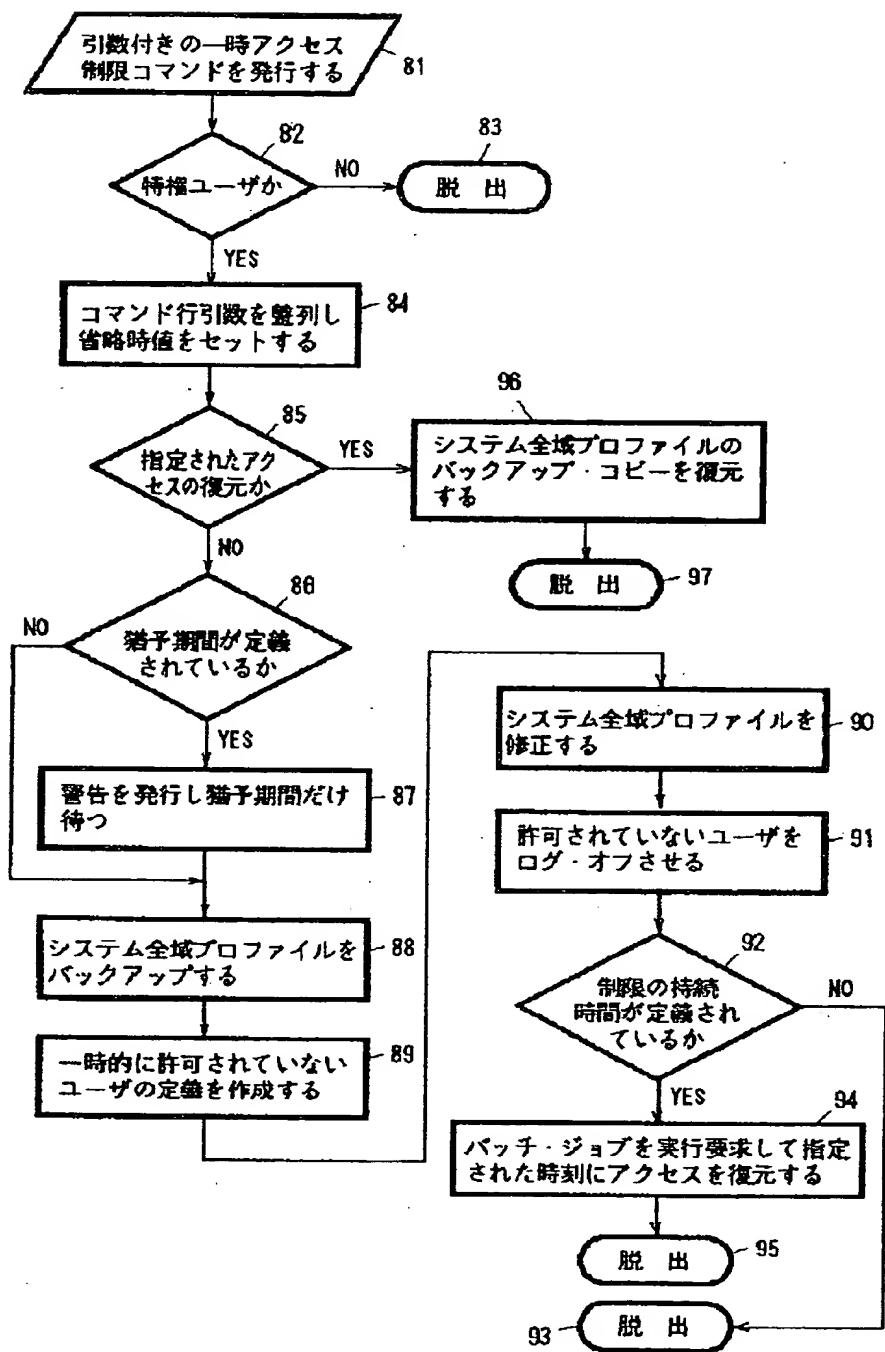
mike:!: 250:100: Mike Taylor : /u/mike : /bin/ksh

</

【図3】



【図 4】



フロントページの続き

(72)発明者 マイケル・ジョージ・テイラー
イギリス エス03 7ビー・ディー ハン
プシャー州サウサンプトン パーク・ゲー
ト セイント・エルモ ダンカン・ロード
(番地なし)